NEURAL NETWORK SCHEMES FOR DATA FUSION AND TRACKING OF MANEUVERING TARGETS

PERFORMANCE REPORT 1
(ONR Grant # N00014-95-1-1224)

Principal Investigator: **Dr. Malur K. Sundareshan**
Professor of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721
Tel. (520) 621-2953; Fax. (520) 621-8076
e-mail:sundareshan@hermes.ece.arizona.edu

ONR Program Manager: **Dr. Rabinder N. Madan**
Surveillance, Communications and Electronics Combat Division
Code 313
Arlington, VA 22217-5660

March 15, 1995

NEURAL NETWORK SCHEMES FOR DATA FUSION AND TRACKING OF MANEUVERING TARGETS

PERFORMANCE REPORT 1
(ONR Grant # N00014-95-1-1224)

Dr. Malur K. Sundareshan, Principal Investigator

ABSTRACT

This is a six-monthly performance report on the ONR sponsored Project "Neural Network Schemes for Data Fusion and Tracking of Maneuvering Targets." This is a new project at the University of Arizona and work on this project was commenced in August 1995. In this report we outline the current status of this project and the work accomplished during the first six months after the project start date. The ability to efficiently fuse information of different forms for facilitating intelligent decision-making is one of the major capabilities of trained multilayer neural networks that is being recognized in the recent times. While development of innovative adaptive control algorithms for nonlinear dynamical plants which attempt to exploit these capabilities seems to be more popular, a corresponding development of nonlinear estimation algorithms using these approaches, particularly for application in target surveillance and guidance operations, has not received similar attention. In this report we describe the capabilities and functionality of neural network algorithms for data fusion and implementation of nonlinear tracking filters. For a discussion of details and for serving as a vehicle for quantitative performance evaluations, the illustrative case of estimating the position and velocity of surveillance targets is considered. Efficient target tracking algorithms that can utilize data from a host of sensing modalities and are capable of reliably tracking even uncooperative targets executing fast and complex maneuvers are of interest in a number of applications. The primary motivation for employing neural networks in these applications comes from the efficiency with which more features extracted from different sensor measurements can be utilized as inputs for estimating target maneuvers. Such an approach results in an overall nonlinear tracking filter which has several advantages over the popular efforts at designing nonlinear estimation algorithms for tracking applications, the principal one being the reduction of mathematical and computational complexities. A system architecture that efficiently integrates the processing capabilities of a trained multilayer neural net with the tracking performance of a Kalman filter is described in this report and the performance of this scheme in a few representative target tracking scenarios is outlined.

TABLE OF CONTENTS

1. INTRODUCTION

Recent advances in sensor technology and distributed computational algorithms, together with demands for escalating operational requirements, have contributed to the growing desire for deploying multiple sensors in military surveillance and tactical mission execution. However, with the growing availability and the need for multisensor operations, there is a corresponding growth in the complexity of these systems mainly arising from the diversity of the types of sensors used and the special problems that may be created such as asynchronous data arrivals and the need to order and correlate these data streams for exploiting the inherent synergy available. Thus, the fundamental question of interest in these scenarios is how to develop an efficient data fusion architecture that facilitates an integrated processing of the large volumes of data arriving at typically high rates to permit the needed decision-making, which in turn facilitates fully utilizing the capabilities of these sensors and realize all possible benefits from their deployment [1].

The ability to efficiently fuse information of different forms is one of the major capabilities of trained neural networks that is being recognized in recent times. While development of innovative adaptive control algorithms for nonlinear dynamical plants which attempt to exploit these capabilities [2] seems to be more popular, a corresponding development of nonlinear estimation algorithms using these approaches, particularly for applications in target surveillance and guidance operations, does not appear to have received a similar level of attention. Principal requirements in these applications are real-time processing capability to deal with the high rate data streams coming in and the ability to rapidly develop appropriate outputs for target detection and for tracking with a high degree of accuracy and reliability. Since neural networks with appropriate training can be endowed with abilities to identify simultaneously multiple correspondences existing between presented data elements and further to utilize these correspondences to produce good global solutions, they seem quite attractive in the development of architectures that meet the requirements stated above. In particular, the primary motivation for employing neural networks in these applications comes from the efficiency with which features extracted from different sensor measurements can be utilized as inputs for developing

the needed estimates on target location and target motion.

This report will give a demonstration of the data fusion abilities of a properly trained neural network. For the sake of providing quantitative performance evaluations for validating this demonstration, we shall consider the application scenario of target tracking. Fig.1a illustrates a schematic where a neural network is employed to fuse the data collected from *m* sensors to develop target state estimate (which could be the position and velocity in the cartesian coordinate system). A variety of sensing devices ranging from radar systems to lasers and optical imaging systems are presently being developed for assisting in high speed target tracking operations executed from both ground-based and airborne platforms. The limitations of using a single sensor in these operations, such as limited accuracy and resolution, and lack of robustness, have motivated the design of tracking systems with multiple sensors which can provide large amounts of useful data to reliably track targets of interest. However, current tracking algorithms usually use information from only one sensor (such as a track-while-scan (TWS) radar) or attempt to combine information from different sensors in an *ad hoc* manner [2-5]. The primary difficulty in these problems stems from the fact that development of tracking algorithms invariably starts with appropriate mathematical models for target motion and one would look for using simple models for target dynamics (linear, for instance) to obtain algorithms which afford simple implementation. Thus, while it is intuitive that using additional data available can facilitate improved tracking decisions, attempting to include this data in the dynamical model of the tracking filter can generally result in significant increases in computational complexity (due to nonlinear relations requiring more complex estimation structures), which may neutralize any performance gains expected from the additional data.

A primary emphasis in our project is placed on quantitatively evaluating the benefits resulting from using a neural network for data fusion while ensuring that the complexity of the tracking algorithm is not increased. Hence, we will select a tracking scenario with a simple implementation structure that uses a linear model support and employs a Kalman filter, and whose performance characteristics are well documented in the literature. We will then include with this scheme one or more additional data forms that are known to lead to complexities due

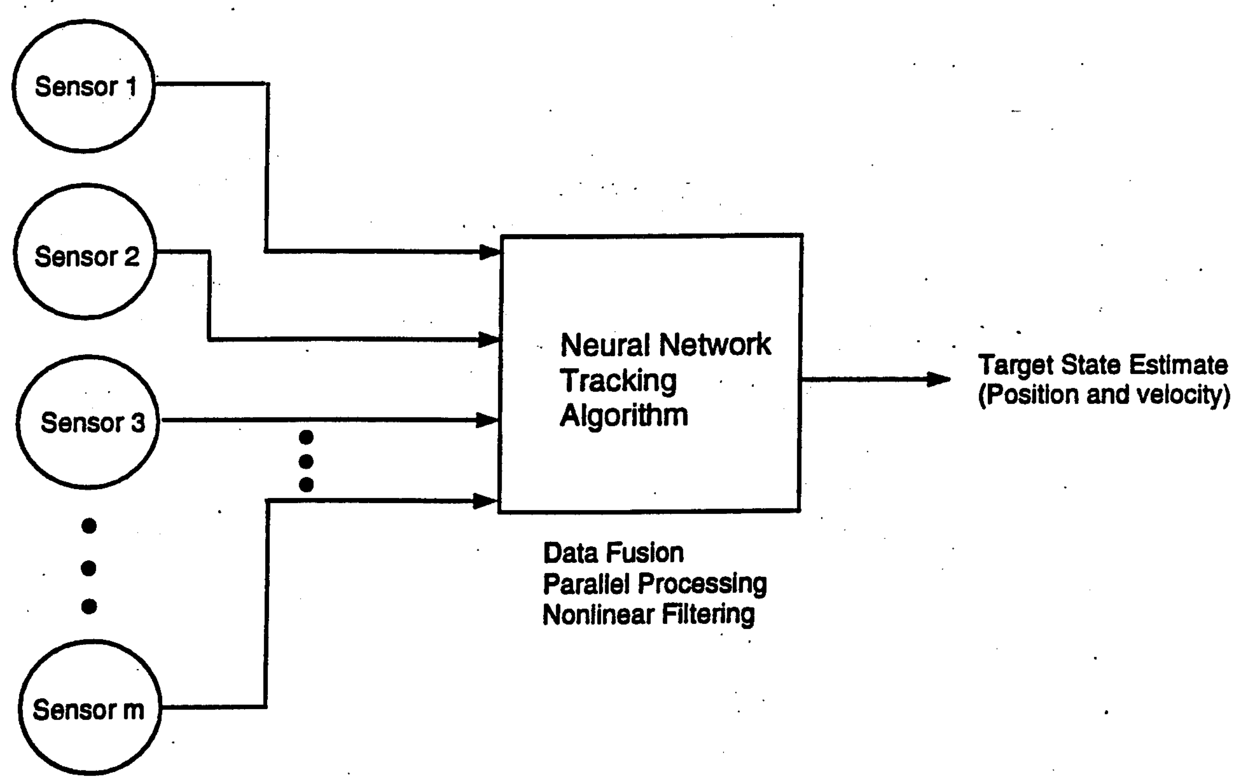


Figure 1a. Neural network processor for multisensor tracking.

to the requirements of support from nonlinear dynamical models and extended Kalman filter algorithms for estimation. To make this comparison study more appealing, we will consider the specific problem of tracking maneuvering targets which offers a considerably challenging environment. It is well known that conventional tracking algorithms based on statistical techniques for processing target returns perform quite efficiently when the target motion does not involve maneuvers. Target maneuvers involving short term accelerations, however, cause a bias in the measurement sequence, which unless compensated results in divergence of the filter in turn leading to a loss of track. Thus, even a single target can be lost by an evasive maneuver and in order to track the target under a maneuvering condition, the tracking window size needs to be expanded. However, as the window increases in size more clutter can enter possibly resulting in false tracks and the loss of true target track. Hence, robust target tracking in these scenarios requires efficient maneuver modeling, *i.e.,* estimation of appropriate "maneuver parameters" that permit a rapid detection of maneuver and correction of target state estimates [5].

The schematic diagram shown in Fig. 1b describes the framework employed for the results presented in this report. A neural network processor works in conjunction with a conventional tracking filter that processes data from a primary tracking sensor (such as a TWS radar). In order to exploit the benefits afforded by the multisensor environment, data collected from various sensors (in addition to the primary sensor) are processed by the neural net for improving the quality of estimates generated by the tracking algorithm. The neural net processor can be tailored to perform the required data fusion for reliably classifying the target maneuver and for estimating the maneuver parameters which are in turn used by the tracking filter. Different neural network architectures can be configured for performing these functions.

The primary motivation for our using a neural network in the specific manner described above is to gain the ability to utilize additional data from a number of sensors in order to realize improved tracking performance while at the same time keeping the computational structure of the tracking filter as simple as possible. These seemingly conflicting goals are possible to achieve owing to the ease with which nonuniform data from different sensors can be utilized as

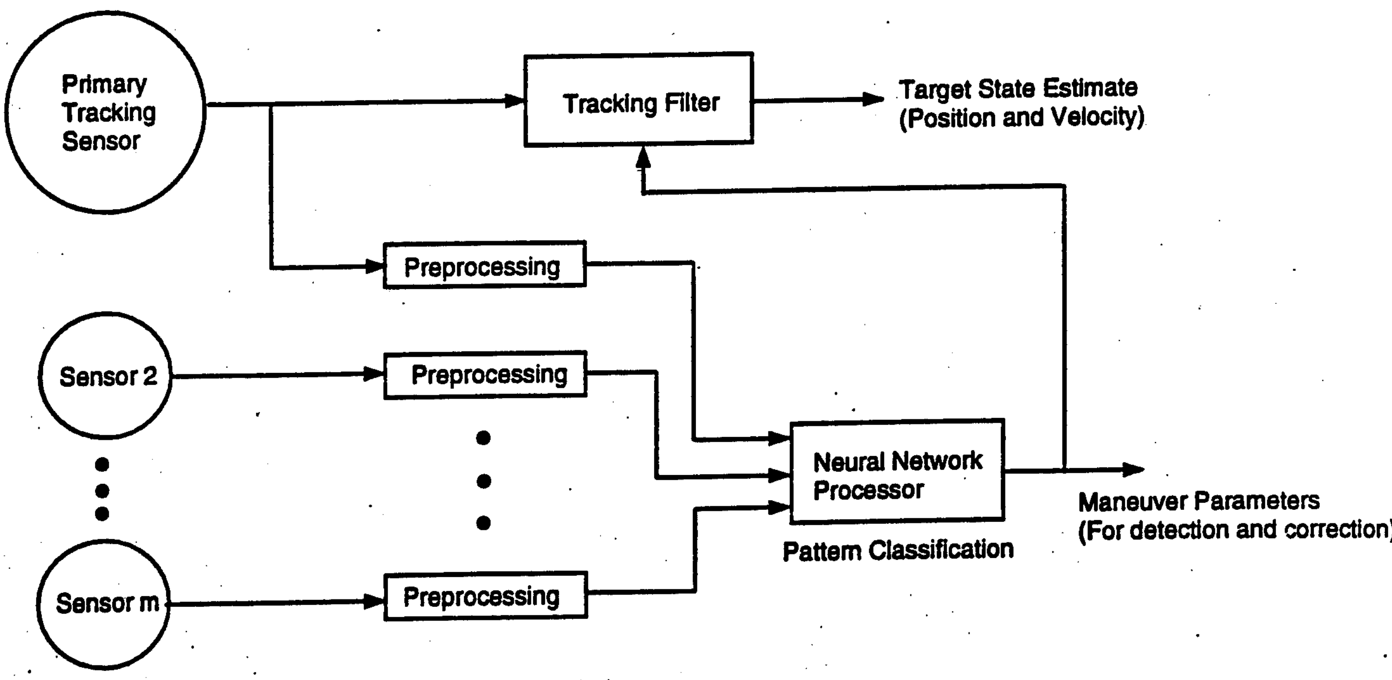


Figure 1b. Neural network for data fusion to assist tracking of target maneuvers

inputs to the neural network either directly or after some preprocessing. Underlying this characteristic is the fact that a neural net processes patterns of data and hence can be trained to process disparate forms of data (radar returns and images, for instance) simultaneously. Furthermore, the parallel processing ability of the neural network can be utilized to perform the maneuver detection and estimation operations rapidly from employing the additional inputs available. Thus, the fundamental idea underlying this approach can be explained as an attempt to enhance the tracking performance realizable from conventional estimation schemes by exploiting the data fusion and parallel processing capabilities of trained neural networks. Performance enhancements can be potentially realized for any estimation scheme one may prefer to employ, whether linear or nonlinear, the idea being to shift the computational burden to the neural network in order to facilitate a more rapid processing of fused data.

The possibility of using a neural network synergistically with existing estimation schemes is perhaps one of the greatest strengths in our approach. Rather than discard an existing tracking algorithm and propose a neural network-based scheme as a total replacement (which may be unacceptable to practitioners due to cost and other considerations), the present approach attempts to tailor the existing features in the problem to a form that could be more efficiently handled by a neural network such that enhanced performance (in terms of track quality, mean track life, etc.) can be realized. Furthermore, some of the required modifications could be effected entirely in software (without any hardware changes), which has obvious implementational benefits.

A fundamental property of a multilayer neural network is that it can provide an arbitrarily close approximation to an unknown nonlinear function from using only the input-output data in numerical form. Hence, the present strategy of synergistically using a neural network with a simple estimation scheme (such as a Kalman filter) can be characterized as resulting in an overall nonlinear tracking filter. However, the present approach differs fundamentally from the currently popular efforts at designing nonlinear filters for tracking applications, such as the approximate nonlinear filters (extended Kalman filter (EKF), iterated EKF, filter designs based on higher order moments and cumulants, etc.) [7] or the exact nonlinear filters (Benes filter, Lie-algebra based designs, etc.) [8-10]. In particular, it overcomes the drawbacks of the latter

approaches, *viz.* high degree of computational complexity and the complexity of mathematics underlying the filter design process. Conceptually, the nonlinear function approximation capability of the neural network enables to simplify the tracking problem by facilitating inclusion of the nonlinear dynamics in the design process while retaining the simplicity of the linear Kalman filter. For the dual problem of control of dynamical plants, application of this idea has become very popular in adaptive control and the efficacy of this approach has been convincingly demonstrated in the control of very complex nonlinear dynamical processes (such as multijointed robotic manipulators [11] and synchronous power generators [12]).

The data fusion capabilities of a trained multilayer neural network and the performance benefits resulting from a system architecture that efficiently integrates these capabilities with the tracking performance of a simple Kalman filter in various maneuvering target tracking scenarios will be delineated in this report. The emphasis in this study is on a proof-of-concept demonstration of the utilization of neural network processing capabilities in reducing the implementational complexity of nonlinear tracking filters. Hence, an illustrative system architecture that efficiently integrates the processing capabilities of a trained multilayer neural net with the tracking performance of a Kalman filter will be described. The performance of this scheme in a few representative target tracking scenarios will be given. A brief outline of the continuing studies on this project will also be given.

2. DESCRIPTION OF TARGET TRACKING ARCHITECTURE

2.1 Problem of Tracking Maneuvering Targets

As noted earlier, target maneuvers involving acceleration inputs that affect target dynamics offer challenging environments for design of tracking systems. The complexities are further compounded if the tracker is required to cope with short-term accelerations where the maneuver time is comparable to the scan time (sampling rate) of the sensor used. In general, the principal requirements of a good tracking system are:

(i) when the target is not maneuvering (*i.e.*, uniform motion of target), position and velocity estimation errors (possibly due to noise) are to be reduced as much as possible:

(ii) a rapid and reliable detection immediately following the on-set of a maneuver is to be ensured:

(iii) the peak estimation error during the maneuver is to be maintained as low as possible (certainly much lower than the unfiltered raw measurements from the sensors).

The development of a target tracking algorithm generally consists of two parts: (a) postulation of a target model (preferably a simple one) that gives an approximate representation of the dynamics of the maneuvering target, and (b) an estimation scheme (typically a Kalman filter or an extended Kalman filter) that provides estimates (preferably optimum in a mean square error (MSE) sense) of the target position and velocity by processing the input data from available sensors. The target model is used as the support for designing the estimation scheme. A number of tracking algorithms have been reported in the literature [3-6] over the years each mainly differing in the way a dynamical system model of the maneuvering target being tracked is obtained.

The general framework employed to model a target maneuver can be described by the dynamical system

$$\dot{x} = f(x,u,v)\,;z_k = h(x_k\,,\,\xi_k) \tag{1}$$

where x denotes the state vector (comprising of target position and velocity variables), u is the input (acceleration) vector, v is the process noise vector (representing possible deviations and uncertainties in the functional representation $f(\cdot)$), z_k denotes the discrete-time measurement vector at time k and ξ_k is the measurement noise vector. While there exist a number of specific algorithms that attempt to model target maneuvers, two approaches that have received a greater degree of popularity in recent times are (i) the multiple model approach, and (ii) the input estimation approach. The former approach centers on the idea of addressing the problem of tracking maneuvering targets through modeling the changes in the function $f(\cdot)$ at the beginning, during and end of a maneuver [14,15]. For illustration, the Interactive Multiple Model algorithm [15] uses different models - a linear model for the uniform motion (no maneuver) and a nonlinear model for the maneuver condition (thus requiring an extended Kalman filter). The Input Estimation approach [16,17], on the other hand, addresses the problem of tracking maneuvering targets through estimation of the input vector u and compensating the Kalman filter output with the estimated input. A specific algorithm following this approach attempts to estimate certain parameters characterizing the maneuver, such as time of occurrence of maneuver and its strength, in a minimum MSE sense, which are then used to compute the state correction required to account for the maneuver. While both of the above approaches have certain strong points and can deliver good tracking performance in specific scenarios, there are some shortcomings as well. Some illustrative ones are the computational complexity and the difficulty in accurately modeling faster-turn and coordinated-turn maneuvers for the multiple model approach, and the need to go several steps in the past to begin correction (which translates into a greater waiting time) and the difficulty in including additional input features (some of which may be readily available) for the input estimation approach.

More advanced schemes that overcome the above limitations and reliably track targets executing even sharp and complex maneuvers under less ideal conditions are useful in practice. These however may necessitate more complex target models, leading to nonlinear estimation algorithms which may not be very attractive due to the mathematical complexity as well as the computational demands placed. Alternate approaches that can deliver improved tracking performance while retaining the simplicity of the filter structure are of particular interest.

2.2 Utilization of Additional Data

An intuitively appealing approach for realizing improved tracking performance under maneuver conditions is to utilize additional data, possibly collected from a diverse set of sensors, in the maneuver detection and state estimation functions. Conventional radars used in surveillance operations do not use information regarding target orientation which is possible to extract from the measurements from imaging sensors (such as a laser radar, infrared sensor or a sensor operating in the visible frequency ranges). It is well known that for maneuvering target tracking applications, while data from radar returns can aid in the determination of target's normal acceleration, determination of the direction of acceleration can be effectively handled from knowledge of target orientation [18]. Even in Air Traffic Control (ATC) applications, it has been argued [15] that the accuracy of tracking algorithms can be significantly improved by including in the estimation process Doppler (range rate) and/or turn rate data (which can be provided by a Mode-S radar).

From a perspective of utilizing the features present in the data for maneuver detection, one may note that most targets of interest in surveillance scenarios are man-made objects which are fairly well structured and result in some specific features in the reflected signal. Furthermore, a target capable of maneuvering generally begins some preliminary actions to prepare for a maneuver, such as banking its wings or lifting the nose. These features, while difficult to extract from the radar returns from a moving target, could be fairly easy to detect from imagery data.

Most of the target tracking algorithms proposed in the literature have the disadvantage of losing the target in short term accelerations. This is due to the fact that these algorithms are typically based on the estimation of some statistical parameter that is used to detect the presence of maneuver and is further used to compute the required corrections to the state estimates. The use of statistical methods for the generation of this parameter often requires that the estimation error over several previous samples is used to ensure that enough samples are considered for a reliable computation. This required waiting time for more samples can result in a total loss of track, since the target can begin a new maneuver.

Thus, in short term accelerations, there is a high potential for the filter not converging on the true track. The only way to resolve this problem appears to be to use more features that are representative of the target maneuver in the estimation process. Intuitively, it is clear that all available data (range, angle, Doppler, etc.) be utilized to the full extent to accurately and rapidly detect and track target maneuvers.

The use of additional data (or more features representative of the maneuver) is not without drawback in existing algorithms. There is no procedure in the current literature that uses data other than the innovation sequence (*i.e.,* position error data) without increasing the dimensionality of the filter. For illustration, the inclusion of turn rate information in a typical multiple model approach [19] increases the filter dimension by one, with the attendant increase in computational complexity. More seriously, any desire to employ additional information (such as range rate (Doppler) and/or features that can be extracted from imaging sensors) within the statistical framework of existing algorithms needs to be tempered with the prospect of having to work with nonlinear motion models for representing target dynamics. The suboptimality of the estimates possible in this case (through extended Kalman filtering approaches) and the increase in algorithmic complexities (with the use of nonlinear estimation approaches) do not generally provide enough encouragement to seek improvements following this route.

An intelligent use of a model independent framework afforded by neural networks can offset these difficulties as will be demonstrated in this paper. Features extracted from other data in addition to the innovation sequence, such as the heading estimate and range rate (which are selected for illustration purposes), can be used to train a neural network for detection of maneuvers and for classifying the intensity of accelerations. Ensuring the feasibility of such a scheme are the two important facts that such training can be conducted entirely off-line (prior to actual deployment in target tracking) and that several very efficient training procedures currently exist. Once properly trained, the neural network can make instantaneously the necessary correlations between the sensor measurements and the maneuver parameters, which are in turn used by the estimation algorithm that performs tracking while retaining the same level of simplicity. Thus, in the present application, the neural network contributes to reducing the potential for track loss by processing more features from available data without increasing the complexity of the tracking algorithm. The data fusion capability underlying this function hence

facilitates a more efficient implementation of the tracking algorithm.

2.3 Neural Network Architecture and Training

Artificial neural networks are emerging as very attractive alternatives to traditional methods (maximum likelihood techniques, nearest-neighbor classification etc.) in the development of computer-based pattern classification algorithms [2,20], since they can learn to perform the required classification without the assumption of probabilistic models for the input patterns. Pattern classifiers are mappings that define partitions of feature space into regions corresponding to class membership. Classification problems that are not linearly separable and require nonlinear decision boundaries can be solved using multilayered neural networks with neurons having nonlinear transfer characteristics. This area has witnessed an explosion of research in the recent past and one of the important results that has come out is based on the celebrated theorem of Kolmogorov. This result states that any continuous nonlinear mapping can be approximated as closely as desired by a multilayered neural network with a feedforward topology and sigmoidal nonlinear functions [21,22].

The basic processing element (neuron) in these function approximating networks has an input-output characteristic which is obtained by forming a weighted sum of the several inputs received and producing an output which is a nonlinear function of this weighted sum, according to the relation

$$y(t) = g\left(\sum_{i=1}^{m} w_i u_i(t)\right) \tag{2}$$

where $u_i(\cdot): \Re \rightarrow \Re, i = 1,2...m$, are the inputs, $y(\cdot): \Re \rightarrow \Re$ is the output, and $w_i \in \Re, i = 1,2...m$, are the weights. $g(\cdot): \Re \rightarrow \Re$ is an appropriately selected nonlinear activation function that satisfies the following conditions:

(i) $x\,g(x) > 0$ for all $x \in \Re$ (first and third quadrant function)

(ii) $\lim_{|x| \rightarrow \infty} g(x) = k\,sgn(x), k > 0$ (saturating function)

(iii) $\frac{g(x_1)}{x_1} \geq \frac{g(x_2)}{x_2}$ for all $|x_1| \leq |x_2|$ (nondecreasing function).

Commonly used activation functions are the sigmoid characteristics (*e.g.*: $g(x) = \tanh(\lambda x)$, or $g(x) = (1+e^{-x})^{-1}$. The processing architecture of an illustrative multilayer feedforward network with one input layer, one output layer and several hidden layers is shown in Fig. 2. In this architecture, the input layer has 3 nodes which merely serve to fan out the incoming inputs to the nodes in the succeeding layer (*viz.*, the first hidden layer) and the output layer has 2 nodes which merely combine the outputs from the nodes in the previous layer (*viz.*, the last hidden layer). The hidden layers have arbitrary numbers of nodes which perform the nonlinear processing according to the rule stated above.

While the processing capabilities of such simple feedforward network architectures are well documented in the literature, more advanced architectures (and analog VLSI implementations) with significantly higher abilities are being developed at present. Among these new architectures, one particular class of networks, *viz.*, neural networks with recurrent connections and dynamical processing elements (neurons), are finding increasing applications in diverse areas. While past efforts at designing such networks have mainly focussed on the steady-state fixed point behavior for such applications as associative memory [23,24] and optimization [25,26], the capabilities offered by these networks for dynamically processing temporal sequences are beginning to be appreciated in the very recent times. Such capabilities render these networks highly valuable for applications to complex dynamical problems such as recognition of continuous time-dependent signals and adaptive control of dynamical systems [11,12]. Application of dynamic recurrent neural networks to problems in surveillance, and target tracking in particular, holds particular interest in providing efficient means for processing temporal data sequences, such as the radar data collected on successive scans of the antenna.

Perhaps the most significant characteristic that enables a neural network to serve as a useful computational device is its learning capability. Implementation of an appropriately tailored learning algorithm, *i.e.,* a rule for adaptive adjustment of the network parameters (specifically the interconnection weights and nonlinear gains), can endow the network with the ability for self-determining the structure to result in a corresponding desired computation. The most popular

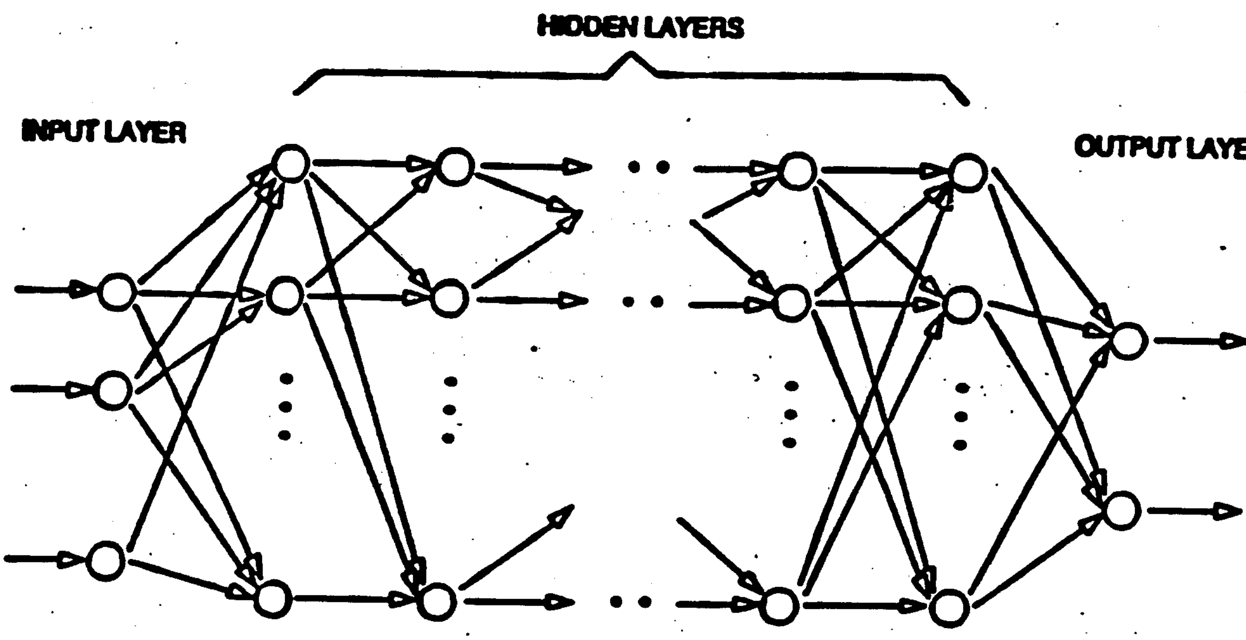


Figure 2. Multilayer feedforward neural network architecture

training scheme used in practice is the error backpropagation algorithm [27]. Several alternative approaches for achieving satisfactory learning capabilities in various constrained data environments are also being developed at present making this topic one of intense current investigation.

Of fundamental importance for a satisfactory training of the neural network is the selection of an appropriate set of input features. By using a sufficient number of features in the training process, it is possible to build very desirable fault tolerance properties (robustness) to the neural network processor. Furthermore, since the network architecture can be appropriately designed to accept as inputs data collected from a number of different sensors, data fusion can be naturally accomplished. Some input preprocessing may however be needed to modify the available data into a form that could be advantageously utilized as network inputs.

Validation of these ideas in radar target surveillance applications has also been obtained from some of our related studies [28] on training neural network processors to perform CFAR detection using radar returns. In this study, the feasibility of using certain statistical parameters (such as the average power over the reference window, median of the sample set, variance of the lagging and leading reference cells, etc.), evaluated from the available data, as inputs to a multilayer neural net was particularly investigated. It was found [28] that a neural network implementation of CFAR detection scheme consistently outperformed conventional cell-averaging approaches in both homogeneous and nonhomogeneous clutter backgrounds, in addition to delivering robust performance (*i.e.,* less degradation in detection probability) when the size of the reference window was reduced or when certain returns from the reference cells were defective. While this study [28] focussed on a different component (*viz.*, detection) of the overall target surveillance operation, these results provide valuable justification of the capability of neural networks for processing data typically measured in surveillance environments and of the benefits resulting from their deployment in enhancing the performance of statistics-based approaches.

2.4 Maneuver Tracking with Data Fusion by Neural Net

Fig. 1b describes the general framework that will be used for illustrating the integration of the data fusion performance of a neural network with a conventional tracking algorithm. We will select the Input Estimation approach [16,17] for discussing the implementation details and for comparing performance results which will be presented in the next section. This selection, it should be emphasized, is for illustrative purposes only and a similar comparison with any other approach can be conducted in an identical manner.

For a brief description, consider a two-dimensional tracking scenario with the dynamics of the maneuvering target described by the linear model

$$x(k+1) = Fx(k) + Gu(k) + v(k) \tag{3}$$

where $x^T(k) = [x(k)\ \dot{x}(k) y(k) \dot{y}(k)]$ is the state vector, $u^T(k) = [u_x(k)\quad u_y(k)]$ is the input vector consisting of the acceleration components in the x and y directions, and $v(k)$ is the process noise. The matrices F and G are given by

$$F = \left[\begin{array}{cc|cc} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{array}\right] \quad \text{and} \quad G = \left[\begin{array}{c|c} \tfrac{1}{2}T^2 & 0 \\ T & 0 \\ \hline 0 & \tfrac{1}{2}T^2 \\ 0 & T \end{array}\right]$$

where T is the time interval between two consecutive measurements. Assuming that the measurement provides the position of the target along the two coordinates, we have the observation sequence

$$z(k) = H\,x(k) + \omega(k) \tag{4}$$

where $z(k)$ is the measurement vector, $\omega(k)$ is the measurement noise and

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Both $v(k)$ and $\omega(k)$ are assumed to be white Gaussian noise sequences with zero means and

covariance matrices Q and R respectively. The two noise processes $v(k)$ and $\omega(k)$ are assumed to be uncorrelated.

In conventional tracking schemes, the received measurements $z(k)$, $k = 1,2,\ldots,$ are processed by a Kalman filter to generate the minimum MSE estimates by implementing the recursive algorithm briefly described in the following steps:

(i) One-step Prediction -

$$\hat{x}(k \mid k\text{-}1) = F\hat{x}(k\text{-}1 \mid k\text{-}1) + Gu(k\text{-}1 \mid k\text{-}1) \tag{5}$$

(ii) Filtering -

$$\hat{x}(k \mid k) = \hat{x}(k \mid k\text{-}1) + K(k)[z(k) - H\hat{x}(k \mid k\text{-}1)] \tag{6}$$

(iii) Gain Computation -

$$K(k) = P(k \mid k\text{-}1)\, H^T[HP(k \mid k\text{-}1)\, H^T + R]^{-1} + Q \tag{7}$$

(iv) Covariance Updating -

$$P\,(k \mid k\text{-}1) = FP(k\text{-}1 \mid k\text{-}1)\, F^T + Q \tag{8}$$

$$P\,(k \mid k) = [I - K(k)\, H]\, P(k \mid k\text{-}1) \tag{9}$$

It may be noted from step (ii) that the filter processes the innovation sequence $\{\tilde{z}(1), \tilde{z}(2), \ldots, \tilde{z}(k), \ldots\}$ where

$$\tilde{z}(k) = z(k) - Hx(k-1)\ ,\ k = 1,2,\ldots\ . \tag{10}$$

When the target is not maneuvering (*i.e.,* when $u(k) = 0$), the mean of the innovation sequence is zero. However, when the target begins to maneuver (*i.e.,* when $u(k) \neq 0$), the mean of $\tilde{z}(k)$ is no longer zero and can be utilized to detect the maneuver.

In the Input Estimation approach [16,17], the build up in the innovation sequence is examined over a detection window by considering several samples in the past. A hypothesis testing method is then used with a specific false alarm rate to detect the maneuver. If a maneuver is detected, the state estimate given by the Kalman filter is corrected by computing a so called "propagation matrix," which involves several multiplications and additions of the state transition

matrix $\Phi(k) = F[I - K(k)H]$. Thus, the maneuver detection and state correction are two distinct steps in this process. The computation of the propagation matrix constitutes a major processing load, which increases with the size of the detection window. Furthermore, the selection of the window size is rather *ad hoc* and any attempts at an optimal selection will require further extensive computations.

The innovation sequence $\{\tilde{z}(1), \tilde{z}(2), \ldots, \tilde{z}(k), \ldots\}$ contains a number of features that are representative of the target maneuver. In the tracking scheme that will be presented here, these features extracted from the innovation data are presented to a trained neural network that classifies the maneuver intensity by developing the estimates of the acceleration components $\hat{u}_x(k)$ and $\hat{u}_y(k)$. For improving the capability of the neural net in developing these estimates, additional features extracted from measurements from other sensors (as indicated in Fig. 1b) can be used. In the presence of a target maneuver, the neural network output is used in the prediction equation (5), which is now modified into

$$\hat{x}(k \mid k\text{-}1) = F\hat{x}(k\text{-}1 \mid k\text{-}1) + G\hat{u}(k\text{-}1 \mid k\text{-}1) \tag{11}$$

until the filter reaches steady state. Appropriate conditions for rapidly reaching steady state can be incorporated into the scheme by appropriate construction of training examples, *i.e.*, by using smaller step sizes of acceleration to generate the training data, as will be discussed in a later section.

It is evident that the scheme described above has several advantages over the Input Estimation approach. First of all, since the neural network outputs the acceleration estimates, no statistics-based test is needed for maneuver detection. The detection of maneuver and state correction are not two separate steps executed one after another, but are implemented together in a continuous fashion. The compensation of bias induced by the maneuver does not require a long waiting time (or a sliding window), since the prediction correction is automatically incorporated in the Kalman filter algorithm through the use of $\hat{u}(k\text{-}1 \mid k\text{-}1)$ (thereby eliminating the need for any additional computations such as the calculation of propagation matrix). Also, any coupling between the two acceleration components (possibly resulting from the transformation of range

and bearing measurements from polar to cartesian coordinates) can be readily included without the requirement of any additional processing (note in contrast that the existing methods generally ignore these coupling effects due to computational complexities). Finally, and perhaps most significantly, a greater reliability in tracking the maneuver is ensured due to the possibility of including measurements from other sensors which can provide features that help detect and classify maneuvers more rapidly and with a greater certainty. Another advantage which may not be immediately apparent is that a more graceful degradation is provided when a false maneuver is detected. This is due to the fact that the network continuously outputs its estimate at each sampling instant and hence, there is no concern with the building up of bias (since both detection and correction take place simultaneously after each sampling period). A particular emphasis should be made here of the data fusion ability of the neural network which is exploited here to result in the benefits stated above with no increase in the computational complexity when compared to that of a simple tracking scheme using a Kalman filter.

Since the underlying processing within the neural network consists of approximating the nonlinear relations existing between the acceleration vector $\boldsymbol{u} = [u_x \quad u_y]^T$ and the features extracted from sensor measurements, the combination of the neural network and the Kalman filter can be characterized as a overall tracking filter which is nonlinear. However, as noted earlier in the Introduction, this approach differs significantly from other mainstream efforts at developing nonlinear tracking filters [8-10] in being implementable without any increase in computational complexity.

3. IMPLEMENTATION DETAILS AND PERFORMANCE EVALUATIONS

3.1 Input Features for Neural Network Training

The efficiency with which target maneuvers can be tracked by the present neural network-based scheme depends on the features used for training the network. This in turn depends on the type of sensors used in the surveillance environment and the type of measurements available for implementing the tracking function. Evidently, depending on whether radar data and/or image-format data are available from the sensors deployed, different sets of features that represent the target accelerations with different degrees of accuracy can be extracted to train the neural network. Selection of appropriate features can be guided by the observation that generally there are three basic entities that help in getting a good estimate of target maneuver. These are: (a) intensity of acceleration, (b) direction of tangential velocity, and (c) initial velocity at the time of acceleration. Each of these entities can be affected by the presence of clutter, however.

Extraction of useful features from raw sensor data not only extends the learning capability of the neural network but also facilitates fusion of diverse data forms by the trained network. Since our emphasis in this study is on a proof-of-concept demonstration, we will select a scenario where data from a TWS radar (primary tracking sensor) is integrated with data from a Doppler radar for the sake of illustrating the details on the feature extraction, the training process and performance evaluation in several maneuver tracking experiments. Furthermore, we will extract two features from the TWS radar data and one feature from the Doppler data to illustrate satisfactory training of the neural network with these features used as inputs. The overall scheme depicting the fusion of Doppler data with the data from TWS radar is shown in Fig. 3. It must be emphasized that the integration of two sensors considered here and the discussion in the remainder of the paper are only for illustrative purposes. Corresponding selections of input features for neural network training and implementation details can be developed in a similar manner when data from other sensors (specifically image-format data) are available for integration [28]. The preprocessing of data to extract the desired features may be preceded by some enhancement and superresolution processing [29,30] if the available resolution in the

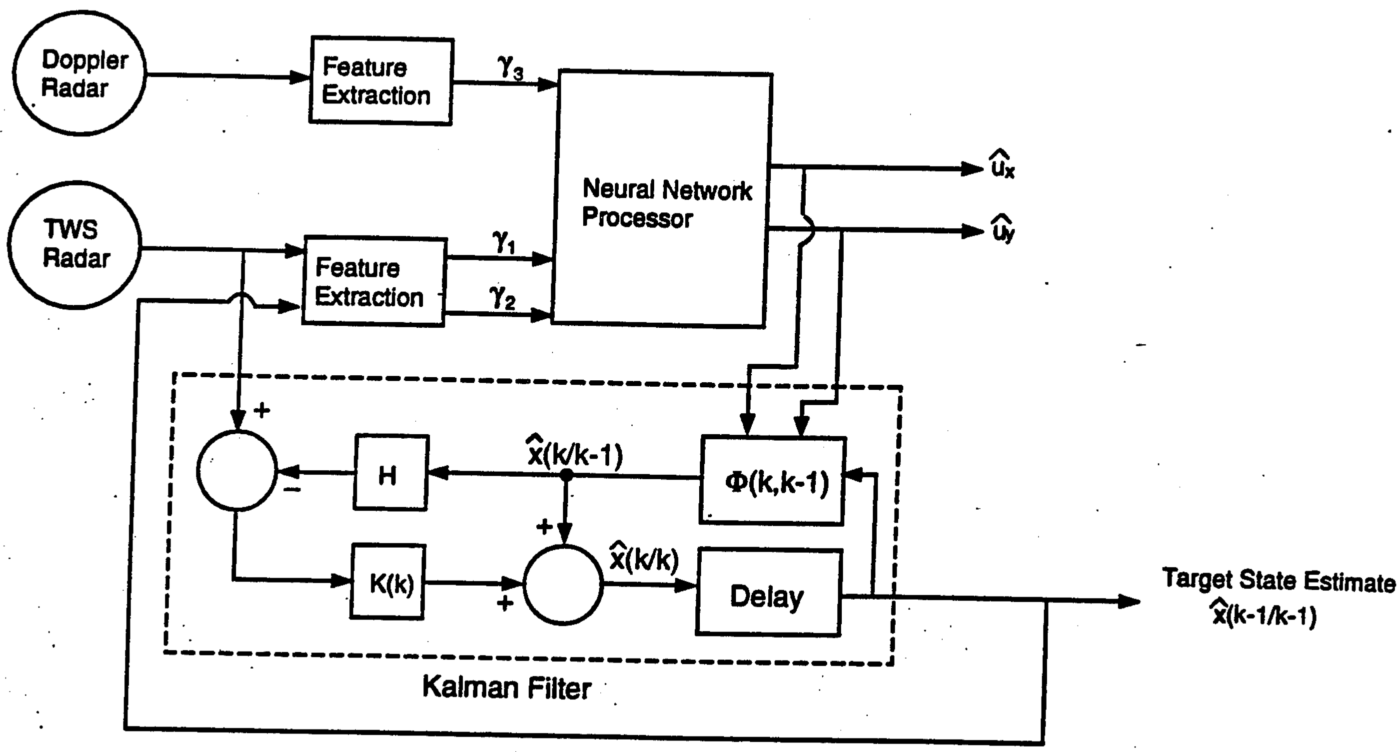


Figure 3. Implementation scenario for fusing data from TWS radar and Doppler radar.

acquired imagery data (particularly in the case of infrared and millimeter wave sensors) is not already enough to obtain the required features.

Feature 1: Normalized position innovation

The first input feature $\nu_1(k)$ used in the present scenario was obtained from the innovation sequence data $\{\tilde{z}(k)\}$, $k = 1,2,....$. The use of this data to obtain inferences on target acceleration levels has been the most popularly used maneuver tracking approach since the work of Magill [31] which proposed using a bank of N parallel filters to match the changes in innovation sequence to acceleration levels. This relation, however, being a nonlinear one, our use of this data for neural network training can be interpreted as learning this nonlinear relation. The signal $\nu_1(k)$is constructed by normalizing the two components of the innovation data with respect to the covariance as

$$\nu_1(k) = \frac{\tilde{z}_x^2(k)}{S_{xx}(k)} + \frac{\tilde{z}_y^2(k)}{S_{yy}(k)} \tag{12}$$

where $\tilde{z}(k) = [\tilde{z}_x(k) \;\; \tilde{z}_y(k)]^T$ defined in (10), and $S_{xx}(k)$ and $S_{yy}(k)$ are the diagonal elements of the covariance matrix

$$S(k) = HP(k \mid k\text{-}1)\, H^T + R$$

which is used for the Kalman filter gain computation in (7).

It may be noted from (12) that we have combined the two components $\tilde{z}_x(k)$ and $\tilde{z}_y(k)$ to produce one feature $\nu_1(k)$. This is merely to keep the number of features in the set of inputs to the neural network a minimum. Of course we could use the two terms on the Right Hand Side of (12) independently, together with some higher level relations among them such as $I_x \cdot I_y$ or I_x / I_y, where I_x and I_y denote these terms. As explained earlier, our interest in the present development is to show that the neural network can be satisfactorily trained to estimate the maneuver if a set of appropriate features are presented as inputs. These features however are nonunique; but they should be descriptive of the maneuver class and also focus on the incremental changes rather than the exact instantaneous values.

In general, attempts at reducing the feature set to represent incremental changes will pay dividends in resulting in a much less training effort. It should also be noted that changes in innovation data are also sensitive to changes in the magnitude of target velocity at the initiation of maneuver. To take this into account, different initial velocities are to be considered in the generation of training set data, which will be explained further in the next section.

Feature 2: Change in Heading

The second input feature $\nu_2(k)$ also comes from the position measurement vector $z(k)$ and is representative of the incremental change in target heading. The importance of this parameter stems from the fact that its first derivative is angular velocity, which gives a sense of the motion of a turning aircraft. Even when the maneuvers are much simpler, use of this parameter is highly beneficial. It is quite well known that tracking systems using heading estimates exhibit a very stable performance even when the sampling interval is increased [32]. The estimation of heading from noisy position measurements is by itself a problem of considerable importance and has received a significant attention. When the target maneuvers, the heading estimate cannot be computed from using the Kalman filter output (*i.e.,* from the estimates $\hat{x}$ and $\hat{y}$) as it is corrupted by bias. Hence, one would like to use the position measurements $z(k) = [x(k) \quad y(k)]^T$ together with some assumptions on the statistics of the noise processes $\omega_x(k)$ and $\omega_y(k)$, which are the two components of the vector $\omega(k)$ in (4).

There are a few approaches in the literature for obtaining the heading estimate from attempting a line fit through the data points as $y(k) = \beta + \alpha x(k)$, of which we briefly cite two most related to the present application. When the variances σ_x and σ_y of $\omega_x(k)$ and $\omega_y(k)$ respectively are known, a maximum likelihood estimate α_{MLE} can be developed from using N data points $\{x_i , y_i\}$, i = 1,2,....N, as

$$\alpha_{MLE} = \frac{\delta + (\delta^2 + 4\sigma p_{xy}^2)^{1/2}}{2p_{xy}} \tag{13}$$

where $\sigma = \sigma_y/\sigma_x$, $\delta = \frac{1}{N-1}\sum_{i=1}^{N} (y_i^2 - x_i^2)$ and $p_{xy} = \frac{1}{N-1}\sum_{i=1}^{N} x_i\, y_i$. One may note that δ is the difference in the variances of the coordinates $\{y_i\}$ and $\{x_i\}$, and p_{xy} is the covariance of these coordinates, and hence, α_{MLE} minimizes the sum of the squared distances (normalized by the variance σ_x and σ_y) from the observed points to the estimated line. An alternate approach, which does not require a knowledge of the variances σ_x and σ_y, employs the "method of least triangles" [33] and obtains the heading estimate

$$\alpha_{LT} = \mu \sqrt{\sum_{i=1}^{N} (y_i - \bar{y})^2 \Big/ \sum_{i=1}^{N} (x_i - \bar{x})^2} \tag{14}$$

where $\mu = \mathrm{sgn}\sum_{i=1}^{N} (y_i - \bar{y})(x_i - \bar{x})$, $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$. The α_{LT} estimate may be preferred due to its lack of requiring a knowledge of σ_x and σ_y, which may need some *a priori* assumptions on the noise processes $\omega_x(k)$ and $\omega_y(k)$. Furthermore, in handling situations where tracking in clutter which may have unknown statistics inside the validation gate is of interest, this will provide obvious benefits. Finally, we recommend using the change in heading as the input feature, *i.e.*, selection of $\nu_2(k)$ as

$$\nu_2(k) = \alpha_{LT}(k) - \alpha_{LT}(k-1) \tag{15}$$

where $\alpha_{LT}(k)$ and $\alpha_{LT}(k-1)$ are the heading estimates computed from using three past data points (*i.e.*, N=3) at the sampling instants k and $(k-1)$ respectively. This feature permits training the neural network for more complex maneuvers (such as coordinated turns) than only accelerations along the direction of tangential velocity vector.

Feature 3: Normalized Doppler Shift

The third input feature $\nu_3(k)$ is extracted from Doppler data and provides a measure of the change in the radial velocity of the maneuvering target. This is computed as the change in Doppler shift normalized by its variance, *i.e.*,

$$\nu_3(k) = \frac{1}{\sigma_{fd}^2} [f_d(k) - f_d(k-1)] \tag{16}$$

where $f_d(i) = \frac{2}{\lambda}\dot{R}(i)$ provides a measure of the radial velocity $\dot{R}\ (i)$ at instant i (λ denotes the wavelength of the transmitted wave). σ_{fd}^2 , the variance of the Doppler shift, is related to the variance of the range rate by

$$\sigma_{fd}^2 = \left(\frac{2}{\lambda}\right)^2 \sigma_{\dot{R}}^2 \tag{17}$$

where $\sigma_{\dot{R}}^2 = \frac{\lambda \cdot (BW)}{4\sqrt{SNR}}$,which can be computed from the bandwidth BW of the Doppler filter and the signal-to-noise ratio SNR [3].

The benefits of using Doppler data in achieving improved tracking performance at the various stages (initialization, estimation of track parameters, plot-to-track association in dense multi-target environments) are quite well known. For purposes of tracking maneuvers, since Doppler shift can be measured within each scan period, this can provide valuable information about the target preparation for a maneuver, and hence can aid in an early detection of maneuver. It must be noted, however, that utilizing this information in conventional tracking schemes requires more complex modifications to the target model (such as using radial velocity as an additional state variable, which not only increases the model dimension but also requires a nonlinear filter to be developed).

A few words on the information learned by the neural network by a presentation of the combined set of features $\{\nu_1\ (k),\ \nu_2(k),\ \nu_3(k)\}$ can give some insight into the training process and the generation of training data. The radial velocity measurement (from the Doppler data) gives only a limited information about the target velocity and may become ineffective when the target path deviates from a radial approach to the radar. Combining the heading information, *i.e.,* using $\nu_2(k)$ in the feature set, ensures that the relevance of Doppler information to the actual target speed is trained to the neural net. Hence, the performance will not degrade severely as the target approaches the radar following a nonradial path (a more quantitative demonstration of this

will be given later by conducting a performance evaluation study in an experiment in Section 3.3). Also of interest to note is that the velocity innovation is more sensitive to the starting and the ending times of the maneuver, while the position innovation is more sensitive to the intensity of the acceleration components. Thus, using the velocity innovation by itself as an indication of maneuver has the drawback of false maneuver detection, since a slight change in the velocity, if not coupled with information on the intensity level, might lead to indicate falsely that a maneuver is detected. This explains clearly why the three parameters together ensure a greater degree of reliability in identifying the maneuver. Besides the improvement in the reliability of decision, the use of more features also has the advantage of requiring less time for completing the decision.

The computations involved in preprocessing of sensor data for evaluating the feature signals $\nu_1(k)$, $\nu_2(k)$ and $\nu_3(k)$ before presenting them to the neural network, as shown in Fig. 3, is merely for simplicity of discussion and to emphasize the fusion property of the neural network which is of central importance in this paper. The feature extraction process can also be made part of the neural network processing with some computational advantages. Specific classes of neural networks (such as Kohonen's self-organizing feature maps [36] and adaptive resonance networks proposed by Carpenter and Grossberg [37]) have been shown to possess unique abilities to assist in this feature.

3.2 Description of Neural Net Architecture and Training

The neural network selected for implementation here is trained off-line with the features $\nu_1(k)$, $\nu_2(k)$ and $\nu_3(k)$ as inputs and the acceleration components u_x and u_y as outputs. Hence, a multilayer network architecture with an input layer comprising of three nodes and an output layer with two nodes is employed. It must be observed that the relations between the components of the acceleration vector $u = [u_x \; u_y]^T$ and those of the feature vector $\nu = [\nu_1 \; \nu_2 \; \nu_3]^T$ are in general complex nonlinear functions that are learned by the neural network from the training examples. The benefits of the model independent framework provided by the network are clearly evident.

The performance of the neural network-based tracking scheme depends on the efficiency of the training process and hence, some care must be given to the generation of training examples. Some details on this process (which resulted in the performance levels that will be depicted in the next section) will now be given. For the sake of conciseness in this description, we will limit this discussion to training the network to learn longitudinal accelerations of the target (*i.e.*, accelerations executed along the tangential velocity direction). It must be emphasized that this is only for purposes of illustrating the details and does not limit the scope of the overall tracking scheme to only these maneuvers. Training the network for any other complex maneuver will simply require generation of more numbers of training examples representative of the specific maneuver class.

Given an expected range of target accelerations, say 0-20m/sec^2, a suitable step size for quantization is selected based on the characteristics of the sensors used. Intuitively, using a large number of quantization levels improves the selectivity of the resulting tracking scheme. However, it must be noted also that concomitant with the reduction in acceleration magnitudes is the fact that clutter will be the major factor in introducing bias in the innovation sequence, which may contribute to a higher rate of false detections. Hence, the resolution chosen should be such that the maximum error equals the standard deviation of the measurements (*i.e.*, σ_R of the radar). Thus, with a maximum acceleration of 20m/sec^2 and a maximum scan time of 10 *sec.* (radar sampling rate), the required precision for the neural network estimate of acceleration components should be within 1m/sec^2 in order to keep the position error below the resolution of the radar. It is simple to note that this error in the acceleration estimate corresponds to 100 meters in the position per scan period. It also corresponds to 10*m/sec.* error in speed, which is typical of an error induced by clutter variations.

Based on an expected range of target velocities, *viz.*, 200*m/sec.* to 700*m/sec.*, a series of training vectors were generated by executing different steps of acceleration with the target moving at different initial velocities. It should be noted that all three of the features used are incremental values which provide the neural network the capability to sense sudden changes in acceleration.

To prepare for the worst case, a Doppler uncertainty of 30*m/sec.* in the velocity innovation is assumed, which results in 15% error for the minimum velocity. Several target trajectories were generated assuming a range of $0 \leq \alpha \leq 45°$ for the heading angle α using appropriate step sizes.

The normalization for computing the feature $\nu_3(k)$ is performed for a worst-case clutter variation (*e.g.* σ_{fd} = 30*m/sec.*). Also a value of $\lambda = 8.57 \times 10^{-3} m$ was employed, which is representative of the wavelengths used in precision tracking systems. It may be noted that for a better Doppler sensitivity, a shorter wavelength must be used. Typically, a millimeterwave radar has a high Doppler sensitivity, *e.g.*, 233.3 *Hz/m/sec.* (that is, for each 1*m/sec.* change in the closing rate $\dot{R}$, the transmitted frequency will shift 233.3 *Hz*), which can detect early changes in target radial velocity. This provides some useful insight into how important this feature is in the pattern recognition of a maneuver.

In the next section we will demonstrate the performance of the present neural network scheme in comparison with some classical techniques that make use of Doppler returns, which will bring out a few interesting observations. For simulation purposes, the range of values for the three input features were established using the parameter values from well known tracking procedures, such as the Probabilistic Data Association Filter (PDAF) [5]. For a brief description of how these ranges are determined, observe that the range of values for the position innovation sequence can be identified from the maximum size of the validation gate [5], which in turn can be evaluated in terms of $S(k)$, the covariance matrix, and a parameter γ obtained from tables of the Chi-square distribution. In this case, the value of γ was selected to be 16 (which corresponds to the 99% probability region). This choice of γ corresponds to a rather heavy clutter environment. As the number of clutter returns increases in the validation gate, the magnitude of the innovation increases and is adjusted by a scale factor q_2 (which depends on the expected number of false measurements and on the detection probability, as described in detail in [5]). From these considerations, the range for $\nu_1(k)$ is obtained as $0.5 \leq \nu_1(k) \leq 2$. Note that the innovation sequence moves toward a smaller value as the clutter increases, that is less probability is assigned to each data originating from the target. As each data falls further away

from the predicted position, the corresponding magnitude of the innovation increases but its Bayesian probability of being originated from target decreases.

Utilizing considerations such as those cited above, the ranges for the three input features were established as,

$$0.5 \leq \nu_1(k) \leq 2 \; ; \; 0.2 \leq \nu_2(k) \leq 1.5 \text{ , and } 0 \leq \nu_3(k) \leq 80,$$

and for the neural network outputs the ranges were developed (as discussed before) as

$$0 \leq \hat{u}_x(k) \leq 20 \quad \text{and} \quad 0 \leq \hat{u}_y(k) \leq 20.$$

$\hat{u}_x$ and $\hat{u}_y$ denote the estimates developed by the neural net for the two acceleration components u_x and u_y . For brevity, details on these are omitted here; they may, however, be found in [34].

The maneuver estimation performance of a neural network with one hidden layer comprising of 14 nodes with nonlinear activation functions $f(z) = (1 + e^{-z})^{-1}$ will be described in the following section. The two output nodes were selected as linear nodes. The network was trained using error backpropagation approach [27] by processing 800 training vectors generated with 20 levels of acceleration covering the range 0-20*m/sec*^{2}, 10 levels of initial velocities covering the range 200-700*m/sec.*, and 4 levels of heading changes. The training was conducted in the batch processing mode using generalized delta rule with momentum [27] for adjusting the weights and with the learning rate selections 0.003 for the hidden layer and 0.018 for the output layer. It may be noted that since innovation data is used for the training, neural network learning of maneuvers takes place when combined with the Kalman filter as in the schematic shown in Fig. 3. The required sensitivity for detection of maneuvers can be built into the neural network by selecting an appropriate threshold for the declaration of maneuvers (for example, for a specified standard deviation of the measurements, say $\sigma_R = 100$, the threshold can be set at an acceleration value of 1*m/sec*^{2}).

3.3 Performance Evaluation

For evaluating the target tracking performance delivered by the present neural network-based

maneuver estimation scheme, and for comparing it with the performance delivered by a few existing algorithms, several simulation exercises that include various types of maneuvers were conducted. These have indicated a consistently superior performance in tracking short term accelerations (where the duration of maneuver τ is comparable to the sampling period T) by the three layer neural network. One may note that in maneuver scenarios where $\tau << T$, the acceleration is too short and a random noise process modeling is usually adequate. Also, for the case when $\tau >> T$, a correlated noise process such as the one generated in Singer's method [6] may be used to model the maneuver and then to compensate for it. The more challenging scenarios, which is where the data fusion abilities of the neural network become highly beneficial, are when the acceleration is not short enough to be considered trivial nor is it long enough to be accurately modeled by purely statistical methods. For reasons of brevity, only a few of the experiments conducted will be briefly outlined. Furthermore, since our intent is to quantify the benefits resulting from fusing data using the neural net, we will restrict ourselves to performance comparisons with some existing maneuver tracking approaches that follow a similar philosophy for detection of maneuver and correction of target state estimates. For ensuring fairness in this comparison, in each experiment 100 independent runs are made and the average of these 100 runs is evaluated.

In each of the experiments reported in this section, the following measurement parameter values were used. The scan period T was selected as 10 *secs*. The standard deviation of measurement error σ_R was assumed to be dependent on the range with a maximum of 100 meters. Therefore, as the target moves away from the radar, σ_R increases to a maximum of 100 meters. The azimuth standard deviation σ_θ was taken to be 0.003 radian. The size of the correlation gate was set to $\gamma=16$.

Experiment #1:

The maneuver scenario in this experiment involves a short duration acceleration with a rather small magnitude. The maneuver is executed by a target which follows a straight trajectory from the initial position of (100*m*, 100*m*) with respect to the radar with an initial speed of 250*m/sec*.

and is radially moving away from the radar with a heading of 45°. The maneuver consists of a sharp acceleration of $5 m/sec^2$ performed at t=40 *sec.* and lasting for only 10 *sec.* (which is one sampling period). The mean filtering errors for the two position coordinates and the x-coordinate velocity (selected for illustration) are shown in Figs. 4a, 4b and 4c, where the performance of the neural network-based scheme (NN) is compared with the Input Estimation (IE) scheme [16,17]. For the IE scheme, a sliding window of length N=2 was used and the correlation gate size (normalized) was set at $\gamma_1=16$ and $\gamma_2=20$ for two consecutive scans. Table 1 gives a concise summary of the track statistics, where track life is defined [5] as the number of consecutive scans during which the target state is estimated with a predefined accuracy (chosen here as position error $\leq 250m$).

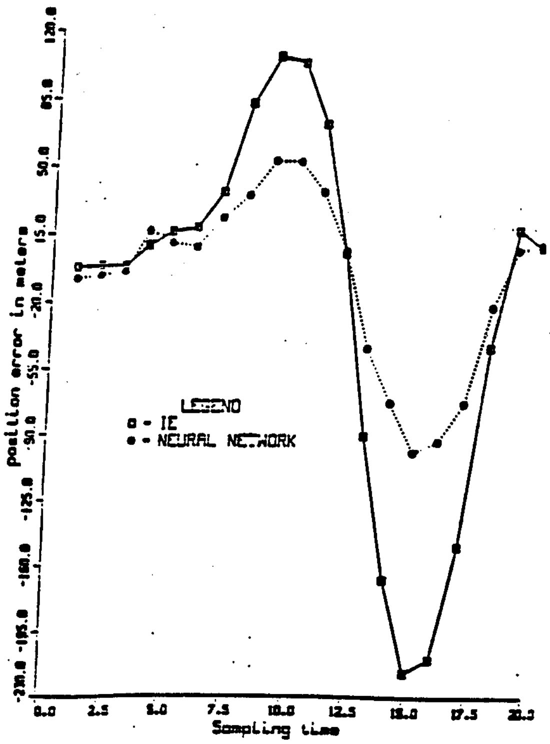
TABLE 1

	% Detection of target along the path	% Detection out of correlation gate	Mean track life in # of scans
NN	100	1.0	19/20
IE	100	1.6	16/20

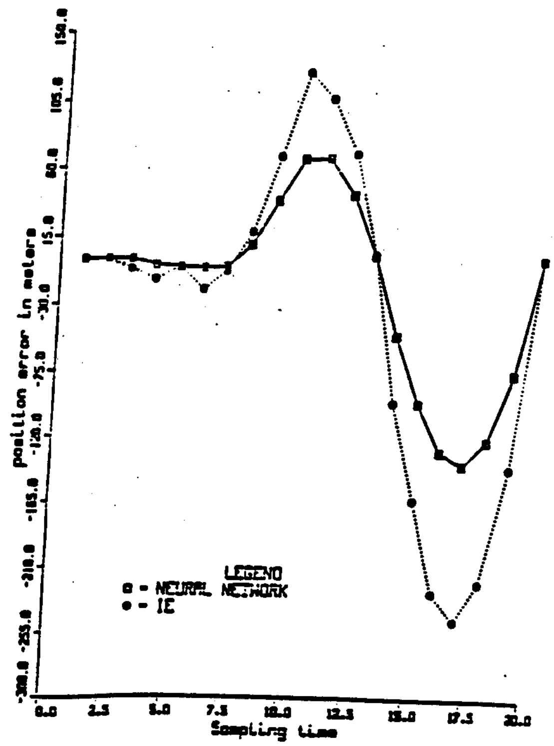
The greater accuracy in tracking by the NN scheme is clearly evident. The performance of the IE scheme can be improved however by selecting a longer sliding window (N>2), with corresponding increase in the required computations. Our objective in this comparison, as noted before, is to demonstrate that the NN scheme is capable of delivering performance levels comparable to existing tracking schemes without increasing the computational requirements. The results of this experiment have confirmed meeting this objective.

Experiment #2:

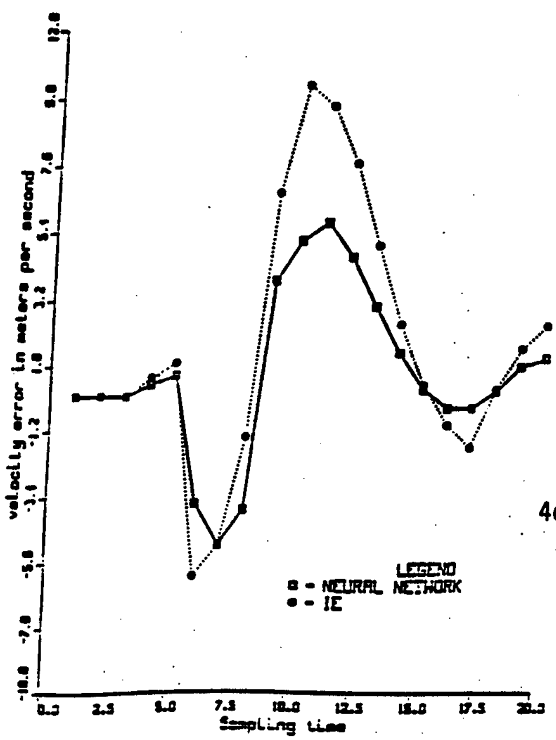
In this experiment, the target starts a maneuver at t=40 *sec.* with an acceleration of $20 m/sec^2$. The target path is the same as in Experiment 1, except that we now add a clutter region which extends between the 10th and 20th scans. The probability of false alarm P_{fa} in this clutter region is assumed to be 0.6 in a gate of $5km$ radius around the predicted target position. The radar



4a. x - coordinate position error



4b. y - coordinate position error



4c. x - coordinate velocity error

Figure 4. Tracking performance of NN scheme in Experiment 1.

coverage is assumed to be 40*km*, and P_{fa} in the clear region is set to 10^{-6}. The measurement uncertainty for the range is assumed to be dependent on the range value and range values of 5,10,15,20,25 and 30 meters were considered. The use of lower values is facilitated by using a high resolution radar with a high Doppler sensitivity of 233 *Hz/m/sec*. The standard deviation of radial velocity was assumed to be 3*m/sec* and the clutter data included the spread of the Doppler clutter spectrum of 30*m/sec*.

The tracking performance delivered by the neural network scheme (NN) in this scenario is shown in Fig. 5 where only the x-coordinate position error is plotted. For comparison, the performance resulting from IE scheme with a window of length N=4 is also plotted. Although this comparison by itself confirms the superior tracking performance of NN scheme, the track statistics summarized in Table 2 more clearly illustrate the benefits. Note that for the NN scheme, an average detection probability P_d of 94.2% was achieved for target detection along the path and only 1.2% of the target data was rejected. The clutter rejection was also considerably better for the NN scheme (37% detection of clutter inside the correlation gate) than for the IE scheme (65% inside the gate). The mean track life for the NN scheme is again higher than for the other scheme.

TABLE 2

	% Detection of target along the path	% Detection out of correlation gate	% Detection of clutter inside the gate	Mean track life in # of scans
NN	94.2	1.2	37	17/20
IE	95.5	2.5	65	14/20

<u>Experiment #3</u>:

The performance improvement is even more dramatic when the target executes a combination of maneuvers and this will be demonstrated in this experiment. The initial trajectory was maintained the same as before with the target initial position at (100*m*, 100*m*) and the initial velocity of 200*m/sec*. The first maneuver takes place at t=60 *sec*. with an acceleration of

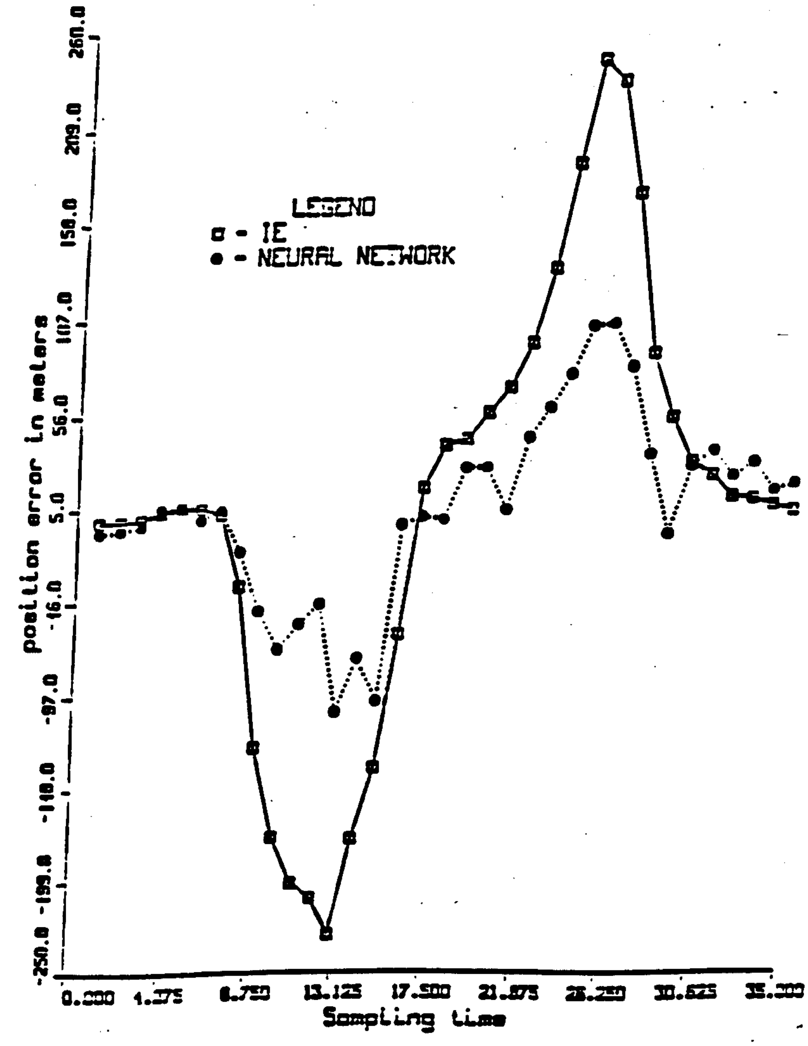


Figure 5. Tracking performance of NN scheme in Experiment 2.

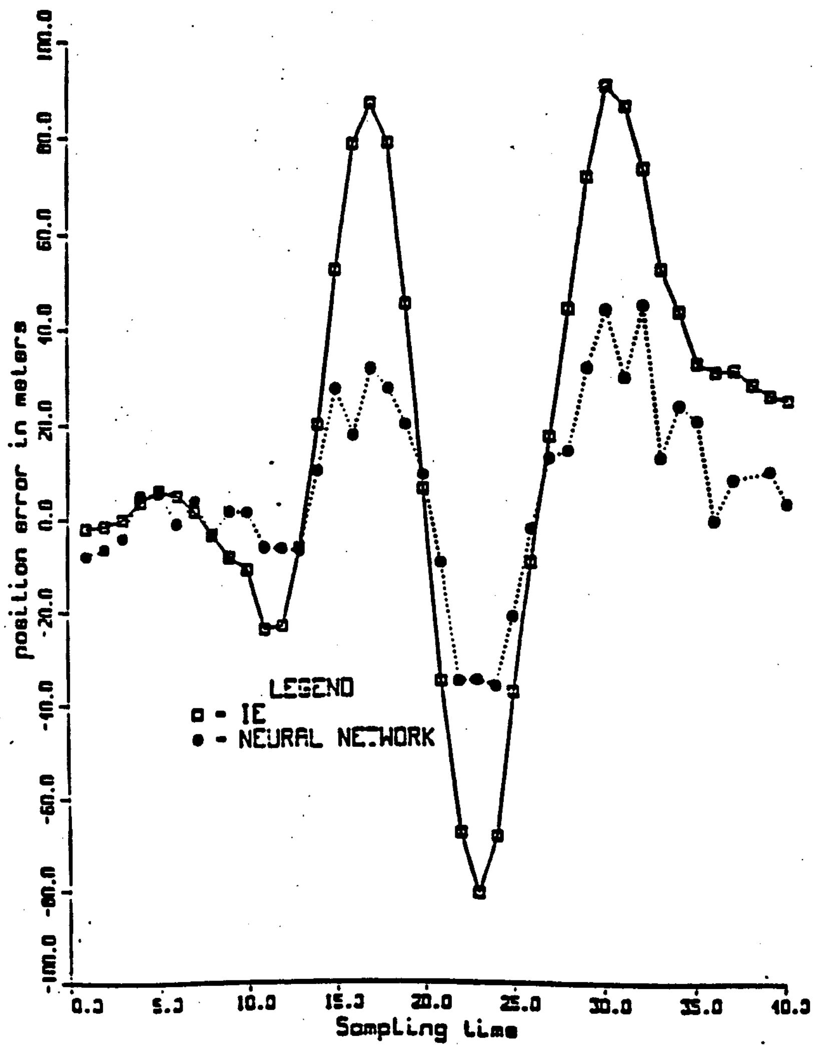
$5m/sec^2$ lasting 10 seconds, which is followed by a second maneuver occurring at t=90 *sec.* (two sampling periods after the first maneuver ends) with an acceleration of $10m/sec^2$ lasting 20 seconds. The measurement uncertainty was retained at the same level as in Experiment 2. The x-coordinate position error plots for the NN and IE (with window length N=3) are compared in Fig. 6a.

To examine the performance degradation when the interval between the two maneuvers is reduced, another experiment was conducted under the same conditions as before except that the second maneuver was commenced at t=80 *sec.* (only one sampling period after the first maneuver ends). The x-coordinate position error plots are again compared in Fig. 6b, which indicates that while the NN scheme retains the same level of performance, the IE scheme suffers a considerable degradation. In particular, notice the sharp peak at scan 25 which is due to the bias that was not fully compensated for from the first maneuver. Thus, an increase in the mean error on top of what is already present due to the first acceleration results. The overall performance is also summarized in Table 3 which indicates a 100% increase in the mean track life for the NN scheme (16/20 against 8/20 for the IE scheme).

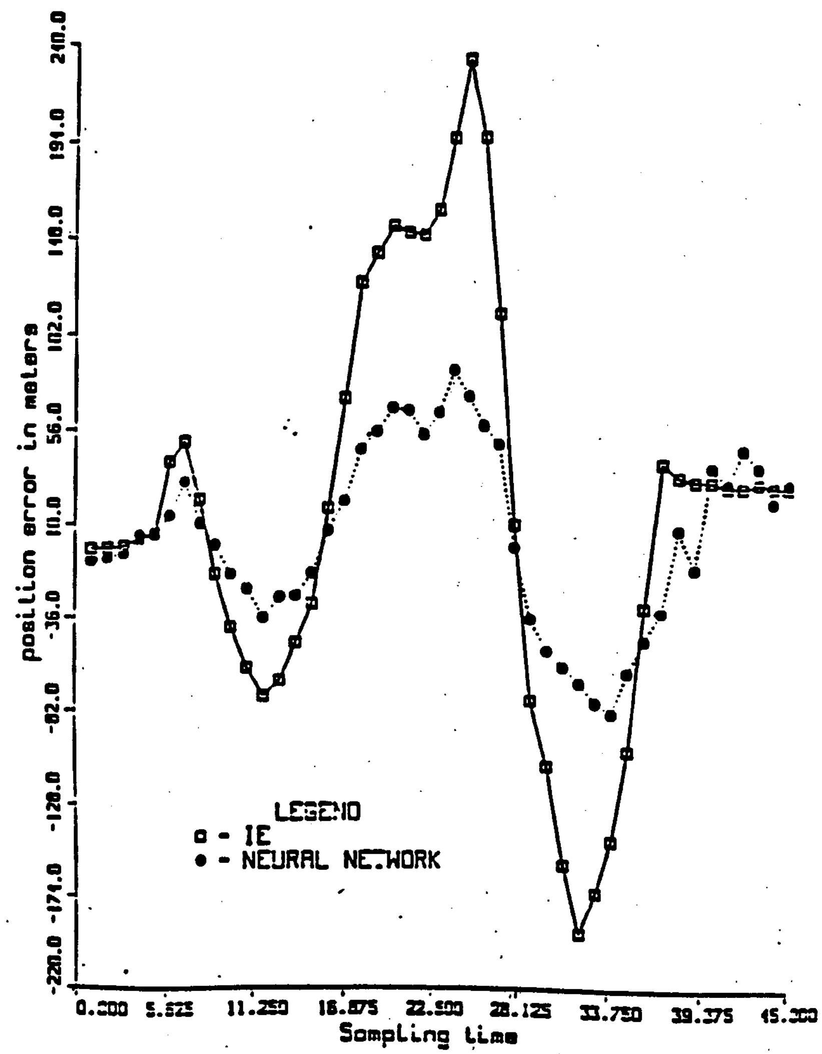
TABLE 3

	% Detection of target along the path	% Detection out of correlation gate	% Detection of clutter inside the gate	Mean track life in # of scans
NN	90.2	2.8	41	16/20
IE	85.0	19.6	63	8/20

It is of particular interest to note that the IE scheme suffers considerable degradation when the interval between the two maneuvers is reduced. In contrast, the NN scheme handles both maneuvers quite well since the first acceleration is well compensated for even before the second one is initiated. With the IE method, however, the second acceleration begins even before the correction due to the first one is fully implemented. The requirement of going back a few periods for computing the propagation matrix is the cause for this shortcoming. The advantage of the NN scheme in this regard is derived from the use of fused data from two sensors, which



6a. Combination of maneuvers two sampling periods apart.



6b. Combination of maneuvers one sampling period apart.

Figure 6. Tracking performance of NN scheme in Experiment 3.

is facilitated by the neural network without incurring any increase in the computational complexity of the filter.

Experiment #4:

In order to investigate whether the neural network has received appropriate training with Doppler data and has learned the nonlinear relation between this data and the target maneuvers, an experiment was conducted to compare the performance of the NN scheme with that of a tracking algorithm that permits incorporating radial velocity measurements. The classical Singer model [6,35], which is based on a correlated noise process that can be represented by an autoregressive (AR) process driven by white noise, provides the best performance under these conditions. The radial velocity information speeds up the initialization phase because it requires only one sample to indicate the target speed instead of two or more samples needed by the position measurements. However, when the target path deviates from the radial approach to the radar, the radial velocity measurement gives only a poor approximation to the actual target velocity and contributes to tracking errors. With the neural network model, however, the target heading information is used as an additional input to train the network. Thus, the relevance of Doppler information to actual target speed is utilized in the training process and hence, the performance does not degrade severely as the target approaches the radar in a nonradial path.

For obtaining a precise quantitative comparison, the following simulation was conducted. A target motion was started from initial position (10*km*, 10*km*) with initial velocity of 350*m/sec.* in a radial trajectory corresponding to the heading angle $\alpha=45°$. Radar scan period was assumed to be 5 seconds. A maneuver was started at t=75 *sec.* which involves a longitudinal acceleration of $20 m/sec^2$ lasting 150 seconds. For the Singer model, the size of the correlation gate was set to 100. Under these conditions, the tracking performance delivered by the neural network scheme is comparable to that of the Singer model as depicted in Fig. 7a which plots the mean position error for the two schemes. This comparison further provides confirmation that the neural network has indeed been trained adequately with Doppler data.

The simulation was then repeated for a nonradial trajectory motion corresponding to the heading angle $\alpha=20°$. The performance from the Singer model in this case degrades considerably, whereas the NN scheme maintains the same level of performance as before. The mean position errors for this case are sketched in Fig. 7b. The underlying reason for the superior performance of the NN scheme is that whereas the radial velocity does not completely reflect the true change in the tangential velocity of the target in the absence of the heading information in the Singer model, the target heading information supplied as an input feature to the neural network facilitates estimating more accurately the velocity changes. This comparison clearly indicates the benefits of including the heading information input (in addition to the other two inputs), which can be realistically handled (without incurring any increase in filter complexity) due to the data fusion abilities of the neural network.

7a. Radial trajectory target motion

7b. Nonradial trajectory motion

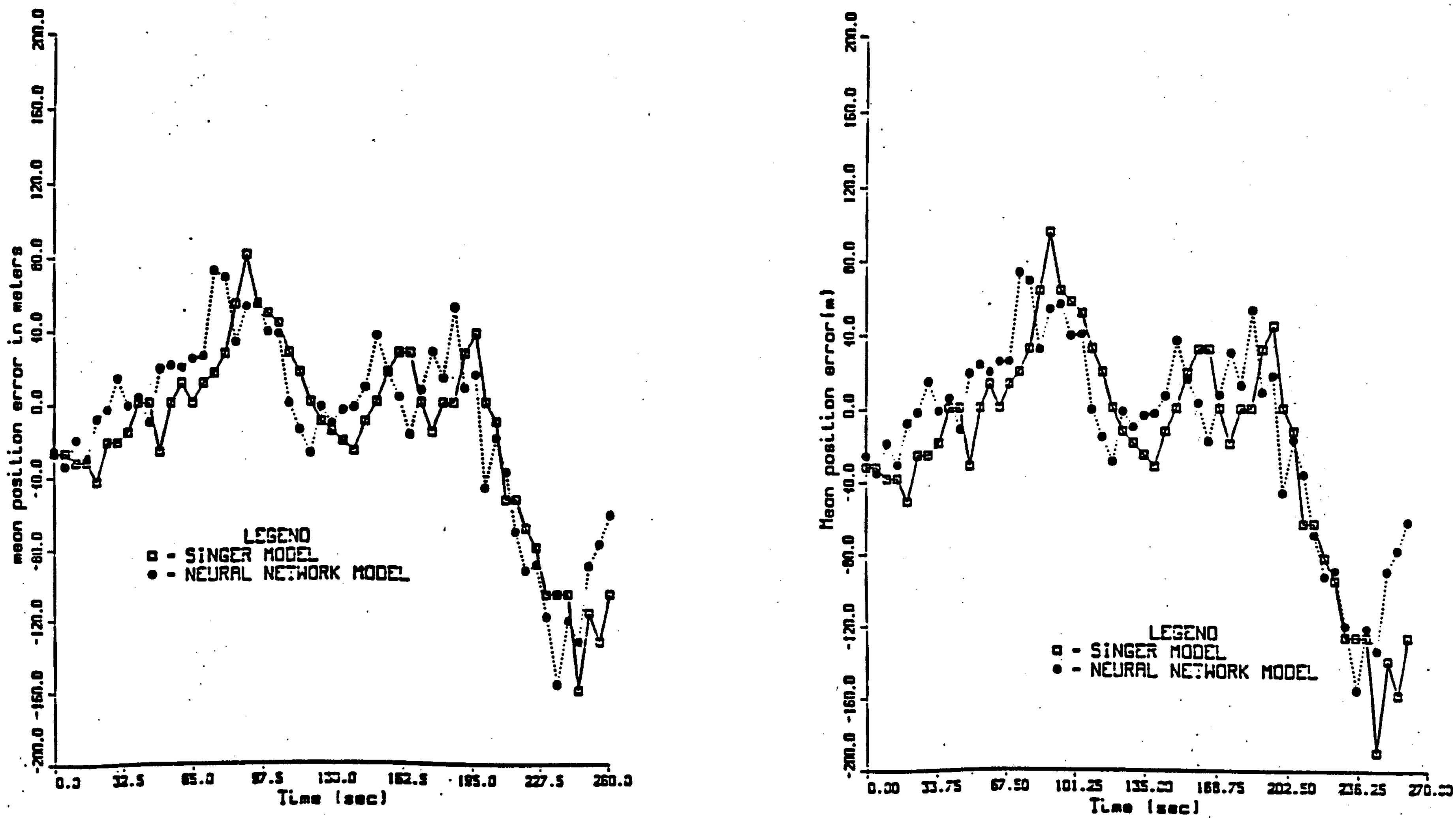


Figure 7. Comparison of NN scheme with Singer model.

4. SUMMARY AND CONTINUING STUDIES

One of the major accomplishments in the studies that have been completed so far in this project is a demonstration of the ability of a neural network to fuse information from different sensors to assist in a simple implementation of target tracking algorithms. The challenging environment posed by the problem of reliably tracking target maneuvers executed in clutter and noise scenarios, from using position measurements from a TWS radar together with data from an additional sensor such as a Doppler radar, was selected to illustrate the development of the tracking architecture and to conduct performance evaluation studies. In particular, it was shown that employing a trained multilayer neural net for processing a set of features extracted from the sensor measurements permits an association to be made rapidly with critical parameters representative of the target maneuver, which in turn facilitates implementation of a conventional estimation scheme (using a Kalman filter) without any increases in computational complexity. The performance evaluation studies conducted in various maneuver tracking scenarios confirm the benefits of using a tracking architecture that includes the deployment of a neural network for data fusion.

The question of how to synergistically integrate the powerful processing abilities of a trained neural net with well known estimation schemes to further improve their performance is at the heart of the present development. Since, at a conceptual level, a neural net can be regarded as an approximator of nonlinear functional relations between input and output data streams, the integration of the neural net with a conventional estimation algorithm can be viewed as an efficient way of implementing an overall nonlinear filter without incurring any attendant mathematical and implementational complexities.

The performance of the neural network architecture developed has been quite encouraging at least in the target tracking scenarios that were tested. We are presently continuing studies on this project to optimize the neural network architecture and to further enhance the target tracking environment by including additional sensor data and conducting performance evaluation studies for more complex target tracking maneuvers. In particular, these studies will focus on the

training of a recurrent neural network to assist in estimating target accelerations more accurately and to exploit utilization of additional features available from data collected from imaging sensors. Performance evaluations comparing the neural network-based schemes with multiple model approaches that have attained particular recognition in the target tracking community are also planned for execution within the next few months.

At a more general level, the development given in this report of a target tracking architecture that uses a neural net for data fusion can be regarded as a special case of employing general learning schemes for enhancing the performance of conventional estimation algorithms. Although tailoring an appropriate neural network generally constitutes an efficient approach for including learning strategies, several other possible approaches, such as learning automata, approximate reasoning, knowledge-based processing and fuzzy logic to name only a few, can also be considered as candidate technologies to contribute towards this overall goal. The development of "Intelligent Estimation" architectures that can be tailored to exploit the powerful data-handling capabilities of these various information processing paradigms to result in improved estimation performance without significant implementation difficulties is thus of considerable potential interest and our work on this project attempts to make a contribution towards this development.

REFERENCES

1. V.G. Comparato, "Fusion - the key to tactical mission success," *Proc SPIE Conf. on Sensor Fusion*, Vol. 931, pp. 2-7, San Diego, CA, 1988.
2. L. Fausett, *Fundamentals of Neural Networks*: Architecture, Algorithms and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1994.
3. A. Farina and S.A. Studer, *Radar Data Processing*, Vol. 1 and Vol. 2, Wiley: Hertfordshire, England, 1985.
4. S.S. Blackman, *Multiple Target Tracking with Radar Applications*, Artech House, Dedham, MA, 1986.
5. Y. Barshalom and T.E. Fortmann, *Tracking and Data Association*, Academic Press, San Diego, CA, 1988.
6. P.L. Bogler, *Radar Principles with Application to Tracking Systems*, Wiley, New York, 1990.
7. A. Gelb, *Applied Optimal Estimation*, MIT Press, 1974.
8. V.E. Benes, "New exact nonlinear filters with large Lie algebras," *Systems and Control Letters,* Vol. 5, pp. 217-221, 1985.
9. F.E. Daum, "Exact finite-dimensional nonlinear filters," *IEEE Trans. on Automatic Control,* Vol. AC-31, No. 7, pp. 616-622, 1986.
10. G. Schmidt, "Designing nonlinear filters based on Daum theory," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 16, pp. 371-376, 1993.
11. A. Karakasoglu, S.I. Sudharsanan and M.K. Sundareshan, "Identification and decentralized adaptive control using dynamical neural networks with application to robotic manipulators," *IEEE Trans. on Neural Networks*, Vol. 4, pp. 919-931, 1993.
12. I. Muhsin, M.K. Sundareshan and A. Karakasoglu, "Adaptive excitation and governor control of synchronous generators using multilayer recurrent neural networks," *Proc. 31st. IEEE Conf. on Decision and Control*, Tucson, AZ, December 1992.
13. M.M. Gupta and N.K. Sinha (Ed.), *Intelligent Control Systems: Concepts and Applications,* IEEE Press, 1995.
14. H.A.P. Blom, "An efficient filter for abruptly changing systems," *Proc. 23rd IEEE Conf. on Decision and Control*, Athens, Greece, December 1986.
15. H.A.P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Automatic Control*, Vol. AC-13, pp. 780-783, 1988.
16. Y.T. Chan, A. Hu and J.B. Plant, "A Kalman filter based tracking scheme with input estimation," *IEEE Trans. on Aerospace and Electronic Systems*, pp. 237-244, 1979.
17. P.L. Bogler, "Tracking a maneuvering target using input estimation," *IEEE Trans. on Aerospace and Electronic Systems*, Vol. AES-23, pp. 298-310, 1987.
18. P.S. Maybeck, R.L. Jensen and D.A. Harnly, "An adaptive extended Kalman filter for target image tracking," *IEEE Trans. on Aerospace and Electronic Systems*, Vol. AES-17, pp. 173-180, 1981.
19. X. Li and Y. Bar-Shalom, "Design of an interactive multiple model algorithm for air

traffic control tracking," *IEEE Trans. on Control Systems Technology*, Vol. 1, pp. 186-194, 1993.

20. S. Haykin, *Neural Networks: A Comprehensive Foundation*, McMillan Press, 1994.
21. K. Hornik, M. Strinchcombe and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, Vol. 2, pp. 359-366, 1989.
22. K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, Vol. 2, pp. 183-192, 1989.
23. J.J. Hopfield, "Neural networks and physical systems with collective computational abilities," *Proc. National Academy of Science*, Vol. 79, pp. 2554-2559, 1982.
24. S.I. Sudharsanan and M.K. Sundareshan, "Equilibrium characteriztion of dynamical neural networks and a systematic synthesis procedure for associative memories," *IEEE Trans. on Neural Networks*, Vol. 2, pp. 509-522, September 1991.
25. J.J. Hopfield and D.W. Tank, "Neural computation of decision optimization problems," *Biological Cybernetics*, Vol. 52, pp. 1-12, 1985.
26. S.I. Sudharsanan and M.K. Sundareshan, "Exponential stability and a systematic synthesis of a neural network for quadratic minimization," *Neural Networks*, Vol. 4, pp. 599-613, 1991.
27. D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning internal representations by error propagation," *Parallel Distributed Processing*, D. Rumelhart and J. McClelland (Eds.), Vol. 1, MIT Press, Cambridge, MA, 1986.
28. H. Pang and M.K. Sundareshan, "Integration of imaging sensor outputs in a radar target tracking environment," Tech. Report ECE-NNT-96-2, ECE Department, Univ. of Arizona, Tucson, AZ, March 1996.
29. T.L. Ji, M.K. Sundareshan and H. Roehrig, "Adaptive image contrast enhancement based on human visual properties," *IEEE Trans. on Medical Imaging*, Vol. 13, pp. 573-587, 1994.
30. M.K. Sundareshan, "Advanced processing techniques for restoration and superresolution of imagery in multispectral seeker environments," Final Report to AFOSR Summer Faculty Research Program, Wright Lab. Armament Directorate, August 1995.
31. D.T. Magill, "Optimal adaptive estimation of sampled stochastic process," *IEEE Trans. on Auto. Control*, Vol. AC-10, pp. 434-439, 1965.
32. S.R. Rogers, "Tracking targets with constant heading and variable speed," *IEEE Trans. on Aerospace and Electronic Systems*, Vol. 26, pp. 543-547, 1990.
33. R.J. Evans, F. Barker and Y.C. Soh, "Maximum likelihood estimation of constant heading trajectories," *Proc. of IEEE Int. Radar Conf.*, London, Ont., pp. 489-493, 1987.
34. F. Amoozegar, "Neural network-based detection and tracking of maneuvering targets in clutter for radar applications," *Ph.D. Dissertation*, Univ. of Arizona, ECE Department, 1994.
35. R.A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Trans. on Aerospace and Electronic Systems*, Vol. AES-6, pp. 473-483, 1970.
36. T. Kohonen, Self-organization and Associative Memory, Springer-Verlag, Berlin, 1984.
37. G. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision Graphics and Image Processing*, Vol. 37, pp. 54-115, 1987.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
March 15, 1996

3. REPORT TYPE AND DATES COVERED
Performance Report #1

4. TITLE AND SUBTITLE
Neural Network Schemes for Data Fusion and Tracking of Maneuvering Targets

5. FUNDING NUMBERS
N00014-95-1-1224

6. AUTHOR(S)
Malur K. Sundareshan

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Electrical and Computer Engineering
College of Engineering and Mines
The University of Arizona
Tucson, AZ 85721

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
Ballston Tower One
800 North Quincy Street
Arlington, VA 22217-5600

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release
Distribution is unlimited

12b. DISTRIBUTION CODE

1

This is a six-monthly performance report on the ONR sponsored Project "Neural Network Schemes for Data Fusion and Tracking of Maneuvering Targets." This is a new project at the University of Arizona and work on this project was commenced in August 1995. In this report we outline the current status of this project and the work accomplished during the first six months after the project start date. The ability to efficiently fuse information of different forms for facilitating intelligent decision-making is one of the major capabilities of trained multilayer neural networks that is being recognized in the recent times. While development of innovative adaptive control algorithms for nonlinear dynamical plants which attempt to exploit these capabilities seems to be more popular, a corresponding development of nonlinear estimation algorithms using these approaches, particularly for application in target surveillance and guidance operations, has not received similar attention. In this report we describe the capabilities and functionality of neural network algorithms for data fusion and implementation of nonlinear tracking filters. For a discussion of details and for serving as a vehicle for quantitative performance evaluations, the illustrative case of estimating the position and velocity of surveillance targets is considered. Efficient target tracking algorithms that can utilize data from a host of sensing modalities and are capable of reliably tracking even uncooperative targets executing fast and complex maneuvers are of interest in a number of applications. The primary motivation for employing neural networks in these applications comes from the efficiency with which more features extracted from different sensor measurements can be utilized as inputs for estimating target maneuvers. Such an approach results in an overall nonlinear tracking filter which has several advantages over the popular efforts at designing nonlinear estimation algorithms for tracking applications, the principal one being the reduction of mathematical and computational complexities. A system architecture that efficiently integrates the processing capabilities of a trained multilayer neural net with the tracking performance of a Kalman filter is described in this report and the performance of this scheme in a few representative target tracking scenarios is outlined.

14. SUBJECT TERMS

15. NUMBER OF PAGES
43

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited